Real-time Software Telemetry Processing System (RT-STPS) User's Guide

Version 6.0

June 2019

DIRECT READOUT
LABORATORY

GODDARD SPACE FLIGHT CENTER
GREENBELT, MARYLAND

Table of Contents

1 General

The NASA Goddard Space Flight Center's (GSFC) Direct Readout Laboratory (DRL), Code 606.3 developed the Real-time Software Telemetry Processing System (RT-STPS) software.

Users must agree to all terms and conditions in the Software Usage Agreement on the DRL Web Portal before downloading this software.

Software and documentation published on the DRL Web Portal may occasionally be updated or modified. The most current versions of DRL software are available at the DRL Web Portal:

https://directreadout.sci.gsfc.nasa.gov

Questions relating to the contents or status of this software and its documentation should be addressed to the DRL via the Contact DRL mechanism at the DRL Web Portal:

https://directreadout.sci.gsfc.nasa.gov/?id=dspContent&cid=66

2 Software Description

RT-STPS ingests raw telemetry data and produces products, including sorted Consultative Committee for Space Data Systems (CCSDS) packets and Virtual Channel Data Units (VCDUs). RT-STPS supports a variety of output formats.

Installed as a server RT-STPS will operate continuously, receiving data from a port or a file and outputting results to files and sockets as specified in a configuration file. A separate interface can be used to invoke RT-STPS from the command line.

The RT-STPS package includes two main utilities: the viewer and the sender. The viewer displays the progress of the server as it runs, and it can be used to load server configuration files. The sender copies a raw data file to the server for processing.

The RT-STPS architecture is depicted in Figure 1. Each server component performs a typical part of the overall CCSDS processing from raw telemetry frames to packets.

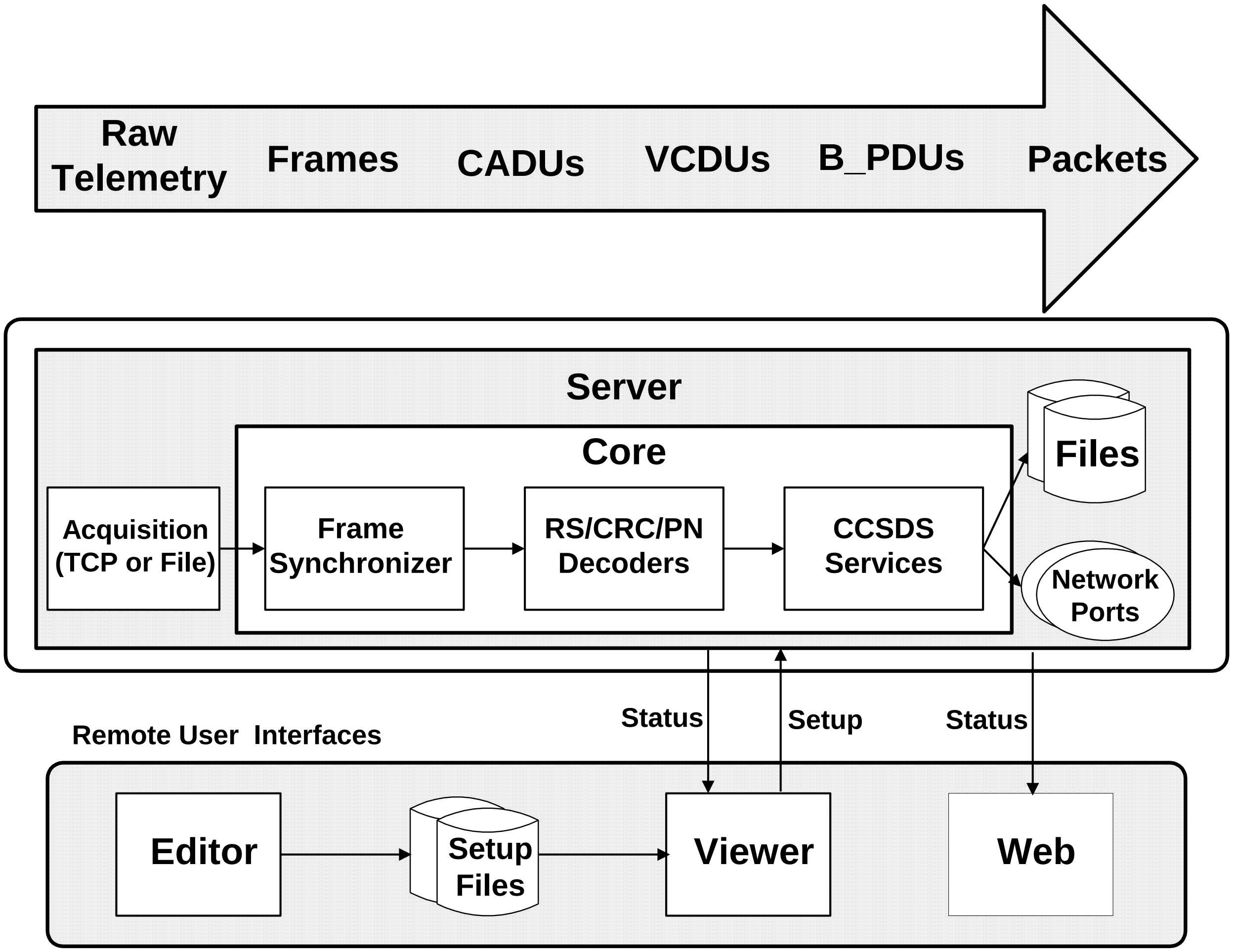


Figure 1. RT-STPS Architecture

3 Software Version

This software package contains RT-STPS Version 6.0. Copyright 1999-2007, United States Government as represented by the Administrator for the National Aeronautics and Space Administration. All Rights Reserved.

Enhancements to Version 6.0 include:

- Support for Joint Polar Satellite System (JPSS-1) processing and the NASA Suomi National Polar-Orbiting Partnership (SNPP) research data model [i.e., Visible Infrared Imaging Radiometer Suite (VIIRS) and Ozone Mapping Profiler Suite (OMPS) Production Data Set (PDS) files for ingest into the International Polar Orbiter Processing Package (IPOPP) v2.6]. JPSS-1 test data is available at: https://is.sci.gsfc.nasa.gov/testdata/jpss1/JPSS-1_Ingest_Verification/
- Implemented XML files, loaded at runtime during Raw Data Record (RDR) processing, which define specifications and metadata for all supported SNPP and JPSS-1 RDR types.

- Updated specifications and metadata for JPSS-1 RDR files.
- Updated configuration files for SNPP and JPSS-1.
- Improved precision for RDR files that cross date boundaries.
- Improved robustness when processing Channel Access Data Unit (CADU) frames containing packet headers and/or data fields that are split across multiple CADU frames.
- Updated leapsec file.

4 Prerequisites

To run this package, you must have Java Development Kit (JDK) (Java 1.6.0_25 or higher) installed on your computer. The JDK is required by some scripts which use the server version of the Java Virtual Machine (JVM), and it is necessary if the RT-STPS distribution will be rebuilt. The bin directory of the JDK must be added to the beginning of the PATH environment variable in the case that the system default Java version is older than 1.6.0_25; placing the bin directory at the beginning of the PATH environment variable ensures use of the correct Java version rather than the system default.

5 Program Inputs and Outputs

RT-STPS ingests raw CCSDS-compliant frames that may be Pseudo-noise (PN)-encoded or Reed-Solomon (RS)-encoded and outputs VCDUs or packets into the following formats:

a) RDR files for Suomi National Polar-Orbiting Partnership (SNPP) and Joint Polar Satellite System 1 (JPSS-1) Mission instruments [Visible Infrared Imaging Radiometer Suite (VIIRS), Advanced Technology Microwave Sounder (ATMS), Cross-track Infrared Sounder (CrIS), Ozone Mapping Profiler Suite (OMPS), and Clouds and Earth’s Radiant Energy System (CERES)];

b) Production Data Set (PDS) (packet file and Construction Record [CSR]) file sets for:

   - Earth Observing System (EOS) Aqua Mission instruments [Atmospheric Infrared Sounder (AIRS), Advanced Microwave Scanning Radiometer (AMSR), Advanced Microwave Sounding Unit (AMSU), CERES, and Moderate Resolution Imaging Spectroradiometer (MODIS)];
   - EOS Terra Mission MODIS instrument;
   - SNPP Mission instruments, as listed in item (a) above;
   - JPSS-1 Mission instruments, as listed in item (a) above.

c) File: header, trailer, and no annotation;

d) Sockets.

6 Installation and Configuration

RT-STPS can be installed on Linux or Microsoft Windows platforms. Installation and configuration instructions for each platform are provided in separate sections: for Linux refer to section 6.1, and for Windows refer to section 6.2. Program operation instructions for both Linux and Windows installations are contained in section 7.

The RT-STPS_6.0_testdata.tar.gz (.zip for Windows) file contains a sample Suomi NPP raw input data file (“rt-stps_npp_testdata.dat”) that may be processed with RT-STPS to test RT-STPS following installation and configuration. Instructions are contained in section 6.1.6 (Linux) and section 6.2.6 (Windows). The “rt-stps_npp_testdata.dat” file will produce RDRs for the Suomi NPP VIIRS, ATMS, CrIS, and OMPS instruments, and PDS file sets for the Suomi NPP Spacecraft, VIIRS, and OMPS instruments. Refer to section 8 for additional information.

6.1 Linux Platform Installation

When installing RT-STPS on a computer where it has been installed previously, be sure to copy any customized configuration files from the previous installation to a safe location so they will not accidently be overwritten during installation.

If you are upgrading from a previous version of RT-STPS, first stop the RT-STPS services, then delete the existing ‘rt-stps’ directory.

6.1.1 Requirements

RT-STPS requires the JDK Java 1.6.0_25 or higher. Download and install the JDK for Linux according to the Oracle/Sun instructions.

The RT-STPS package comes pre-compiled using a Java 1.6 64-bit JDK. A 64-bit version of Linux is required for proper operation.

The bin directory of the JDK must be added to the beginning of the PATH environment variable in the case that the system default Java version is older than 1.6.0_25; placing the bin directory at the beginning of the PATH environment variable ensures use of the correct Java version rather than the system default. Refer to section 6.3 for more details regarding Java requirements.

RT-STPS has been tested on a 64-bit computer with a CentOS Linux 7 X86_64 operating system.

6.1.2 leapsec File Updates

NOTE: Please ensure that all active RT-STPS server instances are stopped before updating the RT-STPS leapsec file. Refer to Section 7, Program Operation, for details on stopping and starting server instances.

RT-STPS requires a leapsec file to calculate the current number of leap seconds to use in time calculations. An up-to-date leapsec file is required for RT-STPS operation. A leapsec file is included in the RT-STPS software package. For correct time calculations, please ensure that an up-to-date leapsec.dat file is located in the root directory of RT-STPS. Leapsec files are available at:

https://is.sci.gsfc.nasa.gov/ancillary/temporal/

The leapsec filename must follow one of the two following formats:

a) leapsec.dat
b) leapsec.YYYYMMDDNN.dat, e.g. leapsec.2013042201.dat

The leapsec file can be updated by running the update_leapsec.sh script, as follows:

```
/home> cd /home/username/rt-stps
/home/username/rt-stps>./bin/internal/update_leapsec.sh
```

6.1.3 Installation

NOTE: These installation instructions assume that the RT-STPS package is installed in '/home/username'.

WARNING: Due to potential resource concerns, RT-STPS should not be installed on a system that is also running an International Polar Orbiter Processing Package (IPOPP) installation.

If the RT-STPS is currently running, stop it by executing the following from the command line:

```
/home> cd /home/username/rt-stps
/home/username/rt-stps>./jsw/bin/rt-stps-server.sh stop
```

If installing RT-STPS on a computer where RT-STPS has been installed previously, first save any customized configuration files to another location before removing the existing 'rt-stps' directory.

Then to install the new package:

1) create a user account (if it does not already exist) under which the server will run;

2) copy the downloaded RT-STPS_6.0.tar.gz file to the desired installation directory;

3) decompress RT-STPS_6.0.tar.gz using the command:

   ```
   $ tar xzvf RT-STPS_6.0.tar.gz
   ```

An 'rt-stps' directory containing the contents of the RT-STPS package should now be installed in the selected installation directory. The RT-STPS user must own and have full read/write permissions to the installation and 'rt-stps' directories. Copy any saved configuration files to the new package location as is appropriate for your system.

Copy the downloaded RT-STPS_6.0_testdata.tar.gz file to the installation directory (same directory as the ‘rt-stps’ directory). Decompress and un-archive the RT-STPS_6.0_testdata.tar.gz file:

```
$ tar –xzf RT-STPS_6.0_testdata.tar.gz
```

6.1.4 Configure RT-STPS

Change to the ‘rt-stps’ directory and run the installation script:

```
/home> cd /home/username/rt-stps
/home/username/rt-stps> ./install.sh
```

In order to make the script executable, it may be necessary to run the command:

```
chmod +x install.sh
```

The following message will appear:

```
Configuring RT-STPS…
```

When configuration is complete, the user will receive the following message:

```
Configuration complete.
```

This message indicates that an RT-STPS package has been installed and configured successfully.

6.1.5 Package Layout

Once the package has been unpacked and installed, the ‘rt-stps’ directory should contain the following contents:

- bin/ – command line applications
- classes/ – compiled java classes directory
- config/ – XML configuration files
- data/ – telemetry data input, and definition files used for RDR processing
- docs/ – javadoc target directory
- images/ – various icon and graphic images
- info/ – directory where auto-generated server info files are stored
- jsw/ – Java Service Wrapper software for Linux
- lib/ – jar files including rt-stps.jar and HDF libraries
- logs/ - contains log files generated by the “server.sh” script
- src/ – source tree
- testdata/ – contains the “rt-stps_npp_testdata.dat” file and test output products
- util/ – contains RT-STPS utilities
- build.sh – Linux build script
- build_javadoc.sh – Linux build javadoc script
- install.sh – installation script for Linux

- leapsec.dat – leapsec file used for accurate time calculations
- rt-stps.dtd – the Data Type Definition (DTD) for the XML configuration files
- rt-stps.policy – policy statement
- SUA Open Source IPOPP GSC-15570-1.pdf – software use agreement
- VERSIONLOG – version info

6.1.6 Testing the Installation

The server sets the ‘rt-stps’ directory as the root directory (RT-STPS files used by the server are searched starting from the root directory), and the configuration files specify that output will go to the ‘../data’ directory. RT-STPS places output into the ‘data’ directory at the same level as the ‘rt-stps’ directory (i.e., ‘../data’).

After testing the installation the user may wish to edit the configuration files to specify a preferred destination directory.

6.1.6.1 Start the Server Manually

To test the installation the server must be started from the command line. Ensure that you are in the ‘rt-stps’ directory and then start the server by issuing the start command to the JSW (Java Service Wrapper):

```
/home> cd /home/username/rt-stps
/home/username/rt-stps>./jsw/bin/rt-stps-server.sh start
```

The JSW captures all output messages to a log file in ‘rt-stps/jsw/logs’. The overall status of the server can also be checked by issuing the status command to the JSW:

```
/home/username/rt-stps>./jsw/bin/rt-stps-server.sh status
```

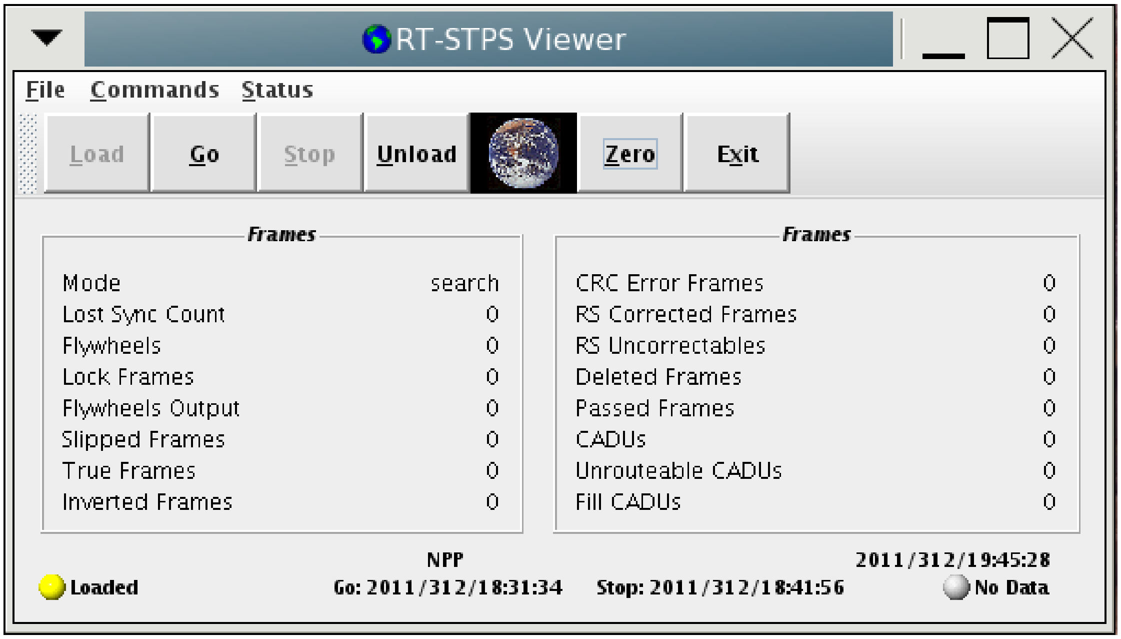
This command will return its current process identifier and a message confirming it is running.

6.1.6.2 Execute the Viewer

Next start the viewer application. The viewer allows the user to load configuration files and to view processing status. From the command line in the ‘rt-stps’ directory enter:

```
/home/username/rt-stps>./bin/viewer.sh &
```

This command will bring up the viewer Graphical User Interface (GUI), depicted in Figure 2. Use the Load button to load the “npp.xml” configuration file from the ‘rt-stps/config’ directory. Use the file dialog to find this directory and file. Once the file loads, click on the Go button.



Frames	
Mode	search
Lost Sync Count	0
Flywheels	0
Lock Frames	0
Flywheels Output	0
Slipped Frames	0
True Frames	0
Inverted Frames	0

Frames	
CRC Error Frames	0
RS Corrected Frames	0
RS Uncorrectables	0
Deleted Frames	0
Passed Frames	0
CADUs	0
Unrouteable CADUs	0
Fill CADUs	0

Loaded

NPP

Go: 2011/312/18:31:34

Stop: 2011/312/18:41:56

2011/312/19:45:28

No Data

Figure 2. RT-STPS Viewer

6.1.6.3 Execute the Sender

Next start the sender application. The sender allows the user to send telemetry data files to the RT-STPS server for processing. It also displays the percentage of the file that has been sent to the server. From the command line in the ‘rt-stps’ directory, enter:

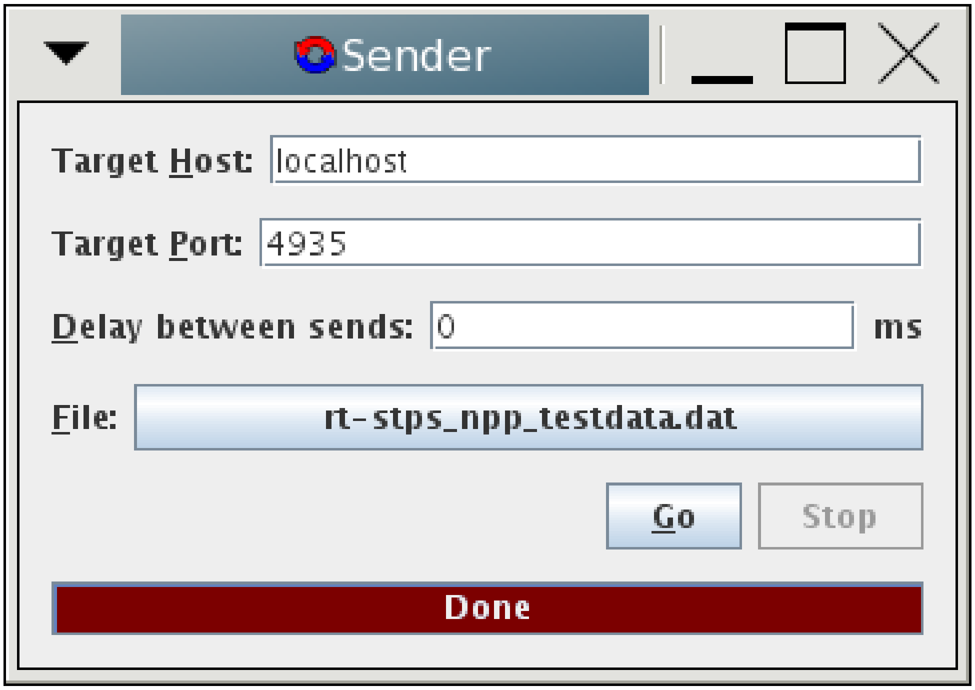
```
/home/username/rt-stps>./bin/sender.sh &
```

This will bring up the sender GUI.

Click on the File button. Use the File dialog to select the “rt-stps_npp_testdata.dat” file from the ‘testdata/input’ directory extracted from the RT-STPS_6.0_testdata.tar.gz file, and click on the Go button to send the file to the server for processing. The sender is depicted in Figure 3.

JPSS-1 test data is available at:
https://is.sci.gsfc.nasa.gov/testdata/jpss1/JPSS-1_Ingest_Verification/

Please refer to the README.txt file at that location for details.



Target Host: localhost

Target Port: 4935

Delay between sends: 0 ms

File: rt-stps_npp_testdata.dat

Figure 3. RT-STPS Sender

6.1.6.4 Inspect the Results

Once processing is complete (the viewer status buttons will show that no more data are being processed, and the sender will show that it is done), list and inspect the contents of the '../data' directory. It should contain RDR and PDS files with filenames similar to the following (the dates of the creation fields will be different):

P1570000AAAAAAAAAAAAAS17264002201000.PDS

P1570000AAAAAAAAAAAAAS17264002201001.PDS

P1570008AAAAAAAAAAAAAS17264002201000.PDS

P1570008AAAAAAAAAAAAAS17264002201001.PDS

P1570011AAAAAAAAAAAAAS17264002201000.PDS

P1570011AAAAAAAAAAAAAS17264002201001.PDS

P1570560AAAAAAAAAAAAAS17264002201000.PDS

P1570560AAAAAAAAAAAAAS17264002201001.PDS

P1570561AAAAAAAAAAAAAS17264002201000.PDS

P1570561AAAAAAAAAAAAAS17264002201001.PDS

P1570826VIIRSSCIENCEAS17264002201000.PDS

P1570826VIIRSSCIENCEAS17264002201001.PDS

RATMS-RNSCA_npp_d20160210_t1613222_e1620501_b00001_c20170921002400312000_all-_dev.h5

RCRIS-RNSCA_npp_d20160210_t1613222_e1620501_b00001_c20170921002400285000_all-_dev.h5

RNSCA-RONPS_npp_d20160210_t1613482_e1620397_b00001_c20170921002400329000_all-_dev.h5

```
RNSCA-ROTCS_npp_d20160210_t1613108_e1620397_b00001_c20170921002400338000_all-_dev.h5
RNSCA-RVIRS_npp_d20160210_t1613068_e1621389_b00001_c20170921002359631000_all-_dev.h5
```

Test output products are available in the ‘rt-stps/testdata/output’ directory. The test output products serve as an indicator of expected program output. Use a comparison utility (such as diff, h5diff, etc.) to compare your output products to those provided in the ‘rt-stps/testdata/output’ directory. Locally generated products may differ slightly from the provided test output products because of differences in file creation times, machine architecture, or operating systems.

6.1.6.5 Stop the Server

Once the test is complete the server may be stopped by issuing the stop command to the JSW:

```
/home/username/rt-stps>./jsw/bin/rt-stps-server.sh stop
```

WARNING: Since the viewer/sender requires the RT-STPS server to be running, it is advised that the sender and viewer be closed prior to stopping the server.

6.1.7 Creating Launchers

Desktop launchers to run the viewer and sender may be created to start these applications. The creation of launchers varies between the different types of Linux distributions and their desktop environments. In all cases the commands described in section 6.1.6.2 and section 6.1.6.3 will apply.

6.1.8 Firewall Configuration

A firewall may in some cases prevent RT-STPS from running correctly, and it may need to be disabled or configured to allow access for certain ports. The server accepts data by default on port 4935. The output port numbers, if any, are defined in the configuration files. The viewer initially connects through port 1099; afterwards, the viewer and server communicate through anonymous ports. The server also uses port 5935 for the alternate server interface.

6.2 Windows Platform Installation

When installing RT-STPS on a computer where RT-STPS has been installed previously, copy any customized configuration files from the previous installation to a safe location so they will not accidentally be overwritten during installation.

If you are upgrading from a previous version of RT-STPS, first stop the RT-STPS services, then delete the existing ‘rt-stps’ directory.

NOTE: This version of RT-STPS for Windows does not support the JSW (Java Service Wrapper).

6.2.1 Requirements

RT-STPS requires the JDK Java 1.6.0_25 or higher. Download and install the JDK for Windows according to the Oracle/Sun instructions. The RT-STPS package is pre-compiled using the Java 1.6.0_25 64-bit JDK with HDF 64-bit.

The bin directory of the JDK must be added to the beginning of the PATH environment variable in the case that the system default Java version is older than 1.6.0_25; placing the bin directory at the beginning of the PATH environment variable ensures use of the correct Java version rather than the system. The easiest way to set the PATH is to use the System Properties dialogue. Refer to section 6.3 for more details regarding Java requirements.

RT-STPS has been tested with Microsoft Windows 7 64-bit.

6.2.2 leapsec File Configuration

NOTE: Please ensure that all active RT-STPS server instances are stopped before updating the RT-STPS leapsec file. Refer to Section 7, Program Operation, for details on stopping and starting server instances.

RT-STPS requires a leapsec file to calculate the current number of leap seconds to use in time calculations. An up-to-date leapsec file is required for RT-STPS operation. A leapsec file is included in the RT-STPS software package. For correct time calculations, please ensure that an up-to-date leapsec.dat file is located in the root directory of RT-STPS. Leapsec files are available at:

https://is.sci.gsfc.nasa.gov/ancillary/temporal/

The leapsec filename must follow one of the two following formats:

a) leapsec.dat
b) leapsec.YYYYMMDDNN.dat, e.g. leapsec.2013042201.dat

6.2.3 Installation

NOTE: These installation instructions assume that the RT-STPS package has been installed in ‘C:\Users\username’.

If RT-STPS is currently running, stop it by executing the command:

```
C:\Users> cd C:\Users\username\rt-stps\bin
C:\Users\username\rt-stps\bin> stop.bat
```

If installing RT-STPS on a system where RT-STPS has been installed previously, first save any customized configuration files to another location before removing the existing ‘rt-stps’ directory.

Then to install the new package:

1) create a user account (if it does not already exist) under which the server will run;

2) copy the downloaded RT-STPS_6.0.zip file to the desired installation directory;

3) decompress the RT-STPS_6.0.zip file directly into the desired installation directory.

An ‘rt-stps’ directory containing the contents of the RT-STPS package should now be installed in the selected installation directory. The RT-STPS user must own and have full read/write permissions to the installation and ‘rt-stps’ directories. Copy any saved configuration files to the new package location as is appropriate for your system.

Copy the downloaded RT-STPS_6.0_testdata.zip file to the installation directory (same directory as the ‘rt-stps’ directory). Decompress and un-archive the RT-STPS_6.0_testdata.zip file directly; if asked to merge directory contents with the existing ‘rt-stps’ directory, select “Yes”.

6.2.4 Configure RT-STPS

Change to the ‘rt-stps’ directory and run the installation script:

```
C:\Users\> cd C:\Users\username\rt-stps
C:\Users\username\rt-stps> install.bat
```

The user will receive the following message:

```
Configuring RT-STPS…
```

When configuration is complete, the user will receive the following message:

```
Configuration complete.
```

These messages indicate that an RT-STPS package has been installed and configured successfully.

6.2.5 Package Layout

Once the package has been unpacked and installed, the ‘rt-stps’ directory should contain the following contents:

- bin\ – command line applications
- classes\ – compiled java classes directory
- config\ – XML configuration files
- data\ – telemetry data input, and definition files used for RDR processing

- docs\ – javadoc target directory
- images\ – various icon and graphic images
- info\ – directory where auto-generated server info files are stored
- lib\ – jar files including rt-stps.jar and HDF libraries
- logs\– location of RT-STPS server log files
- src\ – source tree
- testdata\– contains the “rt-stps_npp_testdata.dat” file and test output products
- build.bat – Windows build script
- build_javadoc.bat – Windows build javadoc script
- install.bat – installation script for Windows
- leapsec.dat – leapsec file used for accurate time calculations
- rt-stps.dtd – the Data Type Definition (DTD) for the XML configuration files
- rt-stps.policy – policy statement
- SUA Open Source IPOPP GSC-15570-1.pdf – software use agreement
- VERSIONLOG – version info

6.2.6 Testing the Installation

By default the server uses ‘..\data’ directory to store processed results. This location is specified in the configuration files in the ‘rt-stps\config’ directory. After testing the installation the user may wish to edit the configuration files to change the destination directory from ‘..\data’ to a preferred location.

6.2.6.1 Start the Server Manually

The JSW is not supported under Windows, so it is necessary to start the server from the command prompt using the provided server script in the ‘rt-stps\bin’ directory. From the command line in the ‘rt-stps’ directory, enter:

```
C:\Users> cd C:\Users\username\rt-stps
C:\Users\username\rt-stps>bin\server.bat
```

The server captures all output to a log file named “rt-stps-server.log” in the ‘rt-stps\logs’ directory. The “rt-stps-server.log” file will appear in the ‘rt-stps\logs’ directory after a configuration file has been loaded with the viewer, as described in section 6.2.6.2.

To check that the server is running, use a command such as “wmic” to display the full command lines and process IDs of all active javaw.exe processes in order to identify the correct javaw.exe process, if there are multiple instances running:

```
C:\Users\username\rt-stps>wmic process where name=“javaw.exe” get
ProcessID,CommandLine
```

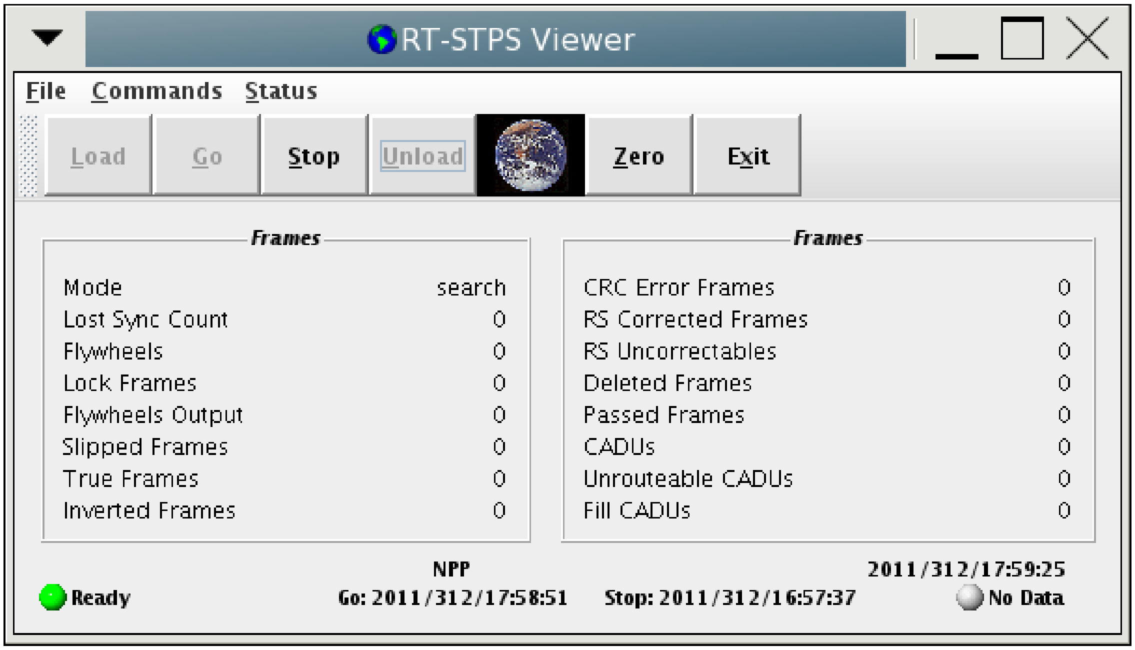
Ensure that the “gov.nasa.gsfc.drl.rtstps.server.TcpServer” process is listed by the above command to verify that the server is running.

6.2.6.2 Execute the Viewer

Next start the viewer application from the command prompt. The viewer allows the user to load configuration files and view processing status. From the command line in the 'rt-stps' directory, enter:

```
C:\Users\username\rt-stps>bin\viewer.bat
```

This command will bring up the viewer Graphical User Interface (GUI), depicted in Figure 4. Use the Load button to load the "npp.xml" configuration file from the 'rt-stps\config' directory. Use the file dialog to find this directory and file. Once the file loads, select the Go button. The viewer GUI is depicted in Figure 4.



```
Frames                                     Frames
Mode                      search           CRC Error Frames          0
Lost Sync Count                0           RS Corrected Frames       0
Flywheels                      0           RS Uncorrectables         0
Lock Frames                    0           Deleted Frames            0
Flywheels Output               0           Passed Frames             0
Slipped Frames                 0           CADUs                     0
True Frames                    0           Unrouteable CADUs         0
Inverted Frames                0           Fill CADUs                0

                    NPP                                  2011/312/17:59:25
Ready   Go: 2011/312/17:58:51   Stop: 2011/312/16:57:37   No Data
```

Figure 4. RT-STPS Viewer

6.2.6.3 Execute the Sender

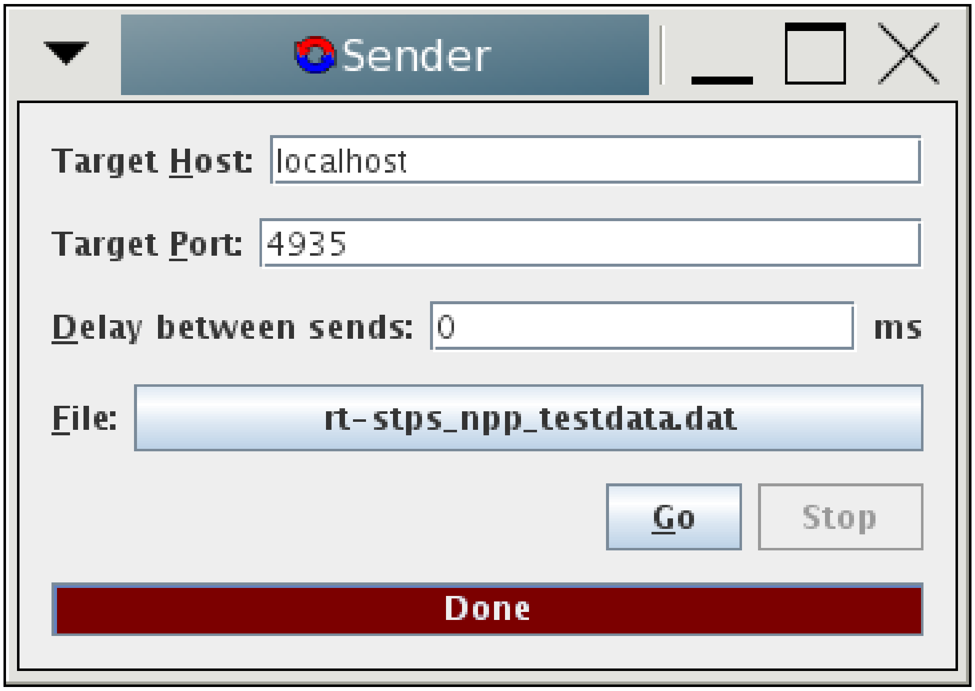
Next start the sender application. The sender allows the user to send telemetry data files to the server for processing. It also displays the percentage of the file that has been sent to the server. From the command line in the 'rt-stps' directory, enter:

```
C:\Users\username\rt-stps>bin\sender.bat
```

This will bring up the sender GUI. On the sender GUI select the File button and use the file dialogue to find and select the "rt-stps_npp_testdata.dat" file in the 'testdata\input' directory extracted from the RT-STPS_6.0_testdata.zip file. Once the file is selected, click on the Go button to send the file to the server for processing. The sender is depicted in Figure 5.

JPSS-1 test data is available at:
https://is.sci.gsfc.nasa.gov/testdata/jpss1/JPSS-1_Ingest_Verification/

Please refer to the README.txt file at that location for details.



Target Host: localhost

Target Port: 4935

Delay between sends: 0 ms

File: rt-stps_npp_testdata.dat

Figure 5. RT-STPS Sender

6.2.6.4 Inspect the Results

Once processing is complete (the viewer status buttons will show that no more data is being processed, and the sender will show that it is Done), inspect the contents of the '..\data' directory. It should contain RDR and PDS files with filenames similar to the following (the dates of the creation fields will be different):

P1570000AAAAAAAAAAAAAS17264002201000.PDS

P1570000AAAAAAAAAAAAAS17264002201001.PDS

P1570008AAAAAAAAAAAAAS17264002201000.PDS

P1570008AAAAAAAAAAAAAS17264002201001.PDS

P1570011AAAAAAAAAAAAAS17264002201000.PDS

P1570011AAAAAAAAAAAAAS17264002201001.PDS

P1570560AAAAAAAAAAAAAS17264002201000.PDS

P1570560AAAAAAAAAAAAAS17264002201001.PDS

P1570561AAAAAAAAAAAAAS17264002201000.PDS

P1570561AAAAAAAAAAAAAS17264002201001.PDS

P1570826VIIRSSCIENCEAS17264002201000.PDS

P1570826VIIRSSCIENCEAS17264002201001.PDS

```
RATMS-RNSCA_npp_d20160210_t1613222_e1620501_b00001_c20170921002400312000_all-_dev.h5
RCRIS-RNSCA_npp_d20160210_t1613222_e1620501_b00001_c20170921002400285000_all-_dev.h5
RNSCA-RONPS_npp_d20160210_t1613482_e1620397_b00001_c20170921002400329000_all-_dev.h5
RNSCA-ROTCS_npp_d20160210_t1613108_e1620397_b00001_c20170921002400338000_all-_dev.h5
RNSCA-RVIRS_npp_d20160210_t1613068_e1621389_b00001_c20170921002359631000_all-_dev.h5
```

Test output products are available in the 'rt-stps\testdata\output' directory. The test output products serve as an indicator of expected program output. Use a comparison utility (such as fc, etc.) to compare your output products to those provided in the 'rt-stps\testdata\output' directory. Locally generated products may differ slightly from the provided test output products because of differences in file creation times, machine architecture, or operating systems.

6.2.6.5 Stop the Server

Once the test is complete the server may be stopped by running the stop script provided in the 'rt-stps\bin' directory. From the command line in the 'rt-stps' directory, enter:

```
C:\Users\username\rt-stps>bin\stop.bat
```

WARNING: Since the RT-STPS viewer/sender requires the RT-STPS server to be running, it is advised that the sender and viewer be stopped prior to stopping the server.

6.2.7 Creating Shortcuts

Create shortcuts to the viewer and sender as follows:

1) In the 'rt-stps\bin' directory, right-click the "viewer.bat" and "sender.bat" files and select "Create Shortcut".

2) Move the shortcuts to the desired location (e.g., the Desktop).

3) Rename the shortcuts "RT-STPS Viewer" and "RT-STPS Sender" respectively.

4) Right-click the shortcuts and select "Properties". Modify the "Start in:" field to be the absolute path of the 'rt-stps' directory. Change the "Run:" selection to "Minimized".

6.2.8 Firewall Configuration

A firewall may in some cases prevent RT-STPS from running correctly, and it may need to be disabled or configured to allow access for certain ports. The server accepts data by default on port 4935. The output port numbers, if any, are defined in the configuration files. The viewer initially connects through port 1099; afterwards, the viewer and server communicate through anonymous ports. The server also uses port 5935 for the alternate server interface.

6.3 Java Setup and Performance Notes (Linux and Windows)

Many of the bash scripts and batch files in the RT-STPS ‘bin’ directory employ the “-server” option to increase performance, as the Sun Java Virtual Machine (JVM) server version may be somewhat faster than the client version of the JVM. By default the JRE comes with the client JVM only, and to use the server JVM, you must install the JDK. If the JDK is installed ensure that is on the PATH before the JRE.

If the “-server” option is used in a script/batch file and the server JVM is not properly installed, the script/batch file will not run properly and terminate with an error.

7 Program Operation

This section includes instructions for both Linux and Windows platforms.

RT-STPS can be started as a server from a command line server script, or from the JSW (Linux only). A configuration file is required to describe the telemetry format specifications and the data outputs for the spacecraft of interest. Typically, the setup configuration file will be loaded prior to each pass. Sample setup files are stored in the ‘rt-stps/config’ directory.

The ‘../data’ directory is the default target for output file results for both the Linux and Windows version of RT-STPS.

A sample raw telemetry file named “rt-stps_npp_testdata.dat” resides in the ‘testdata/input’ directory extracted from the RT-STPS_6.0_testdata.tar.gz (.zip) file. It contains packet data from the VIIRS, ATMS, CrIS, and OMPS instruments aboard the Suomi NPP satellite. A corresponding “npp.xml” configuration file resides in the ‘config’ directory. Use it to produce NASA-model Spacecraft, VIIRS, and OMPS PDS files, and VIIRS, ATMS, CrIS, and OMPS science RDRs. Instructions for testing RT-STPS using these sample files are contained in sections 6.1.6 (Linux) and 6.2.6 (Windows). The RT-STPS package also includes configuration files corresponding to other spacecraft. Refer to section 7.1 for more details.

7.1 Provided Configuration Files

The ‘rt-stps/config’ directory contains pre-made RT-STPS XML configuration files corresponding to all currently-supported spacecraft. The sample XML configuration files are as follows:

a) default.xml – default configuration loaded by RT-STPS if a configuration file is not otherwise specified.

b) aqua.xml – reads Aqua spacecraft telemetry from socket 4935 and writes PDS (packet file and Construction Record [CSR] file pairs) data to files and a port for each of the major instruments aboard the spacecraft.

c) terra.xml – reads Terra spacecraft telemetry from socket 4935 and writes PDS data to files and a port for the MODIS instrument.

d) npp.xml – reads Suomi NPP spacecraft telemetry and will produce Science RDRs for the ATMS, CrIS, OMPS, VIIRS, and CERES instruments, with or without attitude and ephemeris (depending upon their presence in the data). NASA-model Spacecraft, VIIRS, and OMPS PDS files are also generated.

e) jpss1.xml – reads JPSS-1 spacecraft telemetry and will produce Science RDRs for the ATMS, CrIS, OMPS, VIIRS, and CERES instruments, with or without attitude and ephemeris (depending upon their presence in the data). NASA-model Spacecraft, VIIRS, and OMPS PDS files are also generated

7.2 Starting the Server

The server may be started from a command line server script, or from the JSW (Linux only). For Linux, it is strongly recommended to start the server from the JSW.

7.2.1 Command Line Script

Change to the 'rt-stps' directory and run the server script:

<u>Linux</u>

```
/home> cd /home/username/rt-stps
/home/username/rt-stps> ./bin/server.sh &
```

<u>Windows</u>

```
C:\Users> cd C:\Users\username\rt-stps
C:\Users\username\rt-stps>bin\server.bat
```

7.2.2 Java Service Wrapper (Linux Only)

Change to the 'rt-stps' directory and issue the start command to the JSW:

<u>Linux</u>

```
/home> cd /home/username/rt-stps
/home/username/rt-stps>./jsw/bin/rt-stps-server.sh start
```

These commands initiate the server process in the background. When ready, the "Ready to serve" status message is logged to the appropriate location. Refer to section 7.6 for details regarding logging.

7.3 Viewer

The viewer may be run from the command line or as a shortcut from the desktop if this has been configured on your system. To run from the command line, change to the 'rt-stps' directory and execute the viewer script:

Linux

```
/home> cd /home/username/rt-stps
/home/username/rt-stps> ./bin/viewer.sh &
```

Windows

```
C:\Users> cd C:\Users\username\rt-stps
C:\Users\username\rt-stps> bin\viewer.bat
```

The viewer GUI will appear on the screen (see Figure 4). Use the viewer to configure the server and examine its status, as well as to load and unload configuration files.

7.3.1 Viewer Function Summary

Menu Bar
The Menu Bar contains the File, Commands, and Status pull-down menus.

File Menu
The File Menu contains the program Exit item.

Commands Menu
The Commands Menu contains the following commands to configure and run the server:

a) Local Load. Displays a Dialog Box to select and load a configuration file stored on the computer executing the viewer.

b) Remote Load. Displays a Dialog Box to enter the name and load a configuration file stored in the RT-STPS “config/” directory.

c) Go. Starts server data processing.

d) Stop. Halts server data processing.

e) Unload. Removes the current configuration file from the server.

f) Zero Status. Resets the statistics display.

Status Menu
The Status Menu contains menu items to display Path Service Status, Packet Status, the Virtual Channel Status Table, and the Packet Status Table.

Button Bar
The button bar contains buttons linked to the Commands Menu items (see Figure 4).

7.4 Sender

The RT-STPS sender (Figure 3) is used to send a raw data file to the server. The sender may be run from the command line or as a shortcut from the desktop if this has been configured on your system. To run from the command line, changed into the ‘rt-stps’ directory and execute the sender script:

<u>Linux</u>

```
/home> cd /home/username/rt-stps
/home/username/rt-stps> ./bin/sender.sh &
```

<u>Windows</u>

```
C:\Users> cd C:\Users\username\rt-stps
C:\Users\username\rt-stps> bin\sender.bat
```

The Target Host should be “localhost”, the Target Port number should be “4935”, and the delay between sends should be zero. Click on the File button. Use the File dialogue to select the desired raw data file, and click on the Go button to send the file to the server for processing. The server status may be checked using the RT-STPS viewer. When the sender finishes, the server will automatically halt data processing and unload the current configuration file.

7.5 Batch Mode

A batch command performs server processing as a standalone, one-time program. It takes as arguments a configuration file and an input data file containing telemetry frames. The outputs are specified in the configuration file. At the command line, change to the ‘rt-stps’ directory and run the batch script:

<u>Linux</u>

```
/home> cd /home/username/rt-stps
/home/username/rt-stps> ./bin/batch.sh <path-to-config-file> <path-to-data-file>
```

<u>Windows</u>

```
C:\Users> cd C:\Users\username\rt-stps
C:\Users\username\rt-stps>bin\batch.bat <path-to-config-file> <path-to-data-file>
```

NOTE: –D properties are not supported through the scripts.

7.6 Logging

There are two types of RT-STPS output messages: status/event messages, and standard output. RT-STPS logs these messages depending on how the RT-STPS server was started.

Server status/event messages are logged into the “rt-stps-server.log” file in the ‘rt-stps/jsw/logs’ (for the Linux JSW server) or ‘rt-stps/logs’ (for Linux and Windows command line server) directory by default.

Standard output messages generated by the RT-STPS server are only logged if the server is started from the JSW (Linux only). Standard output messages are logged in rolling log files named “rt-stps-server.log.*” in the ‘rt-stps/jsw/logs’ directory, which are automatically managed by the JSW.

Additional log options are possible. Refer to section 10 for more information on server configuration details.

7.7 Stopping the Server

The server may be stopped from a command line script or from the JSW (Linux only), depending on how it was started. Because the sender and viewer require the RT-STPS server to be running, it is advised that the sender and viewer be stopped prior to stopping the server.

7.7.1 Command Line Script

If the server was started using the command line server script, change to the ‘rt-stps’ directory and run the stop script:

Linux

```
/home> cd /home/username/rt-stps
/home/username/rt-stps> ./bin/stop.sh
```

Windows

```
C:\Users> cd C:\Users\username\rt-stps
C:\Users\username\rt-stps> bin\stop.bat
```

7.7.2 Java Service Wrapper (Linux Only)

If the server was started from the JSW, change to the ‘rt-stps’ directory and issue the stop command to the JSW:

Linux

```
/home> cd /home/username/rt-stps
/home/username/rt-stps>./jsw/bin/rt-stps-server.sh stop
```

7.8 Rebuilding

If it is necessary to rebuild the RT-STPS distribution, stop the server (refer to section 7.7) then run the build scripts:

Linux

```
/home> cd /home/username/rt-stps
/home/username/rt-stps>./build.sh
```

Windows

```
C:\Users> cd c:\Users\username\rt-stps
C:\Users\username\rt-stps>build.bat
```

This command will compile the Java source files and create the “rt-stps.jar” file in the ‘lib’ directory.

Warning for Linux users: The build.sh script calls the Java Remote Method Invocation Compiler (RMIC) “rmic” program to build the remote method invocation portion of RT-STPS. The GNU version of RMIC cannot be used with the RT-STPS build script. Ensure that the Java rmic program is before the GNU RMIC program in the system PATH, so the build.sh file will execute properly. Alternatively, edit the build.sh script and hard-code the absolute path to Java rmic.

Warning for Windows users: The build.bat script calls the “rmic.exe” program to build the remote method invocation portion of RT-STPS. However, if Cygwin is installed on your system, the distribution may include a program of the same name. The Cygwin version of RMIC cannot be used with the RT-STPS build batch file. Ensure that the Java rmic.exe program is before the Cygwin RMIC program in the system PATH, so that the build.bat file will execute properly. Alternatively, edit the build.bat file and hard-code the absolute path to Java rmic.exe.

8 Raw Data Record (RDR) Creation

8.1 Supported RDRs

RT-STPS Version 6.0 supports the creation of Raw Data Records (RDRs) for the Suomi NPP and JPSS-1 VIIRS, ATMS, CrIS, OMPS, and CERES instruments. Attitude and ephemeris packets included in the input produce the corresponding Spacecraft Diary with each RDR.

8.2 Test Data

The RT-STPS_6.0_testdata.tar.gz (.zip for Windows) file contains a sample Suomi NPP raw input data file named “rt-stps_npp_testdata.dat”.

This file may be processed with RT-STPS to produce RDRs for the Suomi NPP VIIRS, ATMS, CrIS, and OMPS instruments.

8.3 Suomi NPP and JPSS-1 Configuration File for RDRs

RT-STPS includes the “npp.xml” and “jpss1.xml” configuration files that define everything necessary to create the Suomi NPP and JPSS-1 VIIRS, ATMS, CrIS, OMPS, and CERES

RDRs respectively, as well as to output VIIRS science packets to a socket interface. The "npp.xml" and "jpss1.xml" configuration files can be used as the basis to create other RDR configurations of interest (e.g., a configuration file dedicated solely to VIIRS Science RDRs, among others). The RDR creation process is depicted in Figure 6.

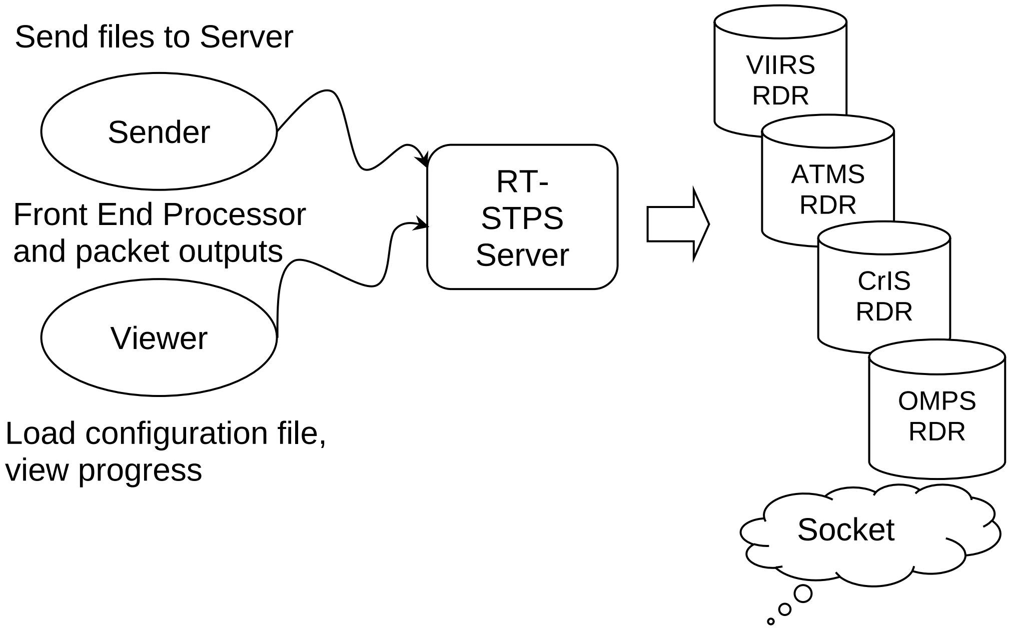


Figure 6. RDR Creation Process

8.4 Processing the Test Data

Decompress and un-archive the RT-STPS_6.0_testdata.tar.gz (.zip) file. Refer to section 6.1.6.1 (Linux) and section 6.2.6.1 (Windows).

Use the viewer to load the configuration file. Refer to section 6.1.6.2 (Linux) and section 6.2.6.2 (Windows) for instructions to execute the viewer. Use the sender to send the 'rt-stps_npp_testdata.dat' file to the server for processing. Refer to section 6.1.6.3 (Linux) and section 6.2.6.3 (Windows) for instructions to execute the sender.

On both Linux and Windows installations, the server uses the '../data' directory by default to store processed results. Refer to section 6.1.6.4 (Linux) and section 6.2.6.4 (Windows) for additional details.

8.5 RDR Processing Status

When RT-STPS is run in server mode, the RDR processing module updates some status information as it creates granules for the RDR of interest. The status information may be viewed using the RT-STPS viewer, or by using the getstatus script (refer to section 9.1). In batch mode, this information is output to the console directly. Refer to section 7.5 for instructions to run RT-STPS in batch mode.

8.6 Expected Output

Once processing is complete, the viewer status buttons will show that no more data is being processed, and the sender will show that it is done. The '../data/'directory should contain the generated RDR files. Refer to section 6.1.6.4 (Linux) and section 6.2.6.4 (Windows) for more details regarding the expected output when processing the "rt-stps_npp_testdata.dat" raw data file.

8.7 Reference Sources

Details on mission data specifications are contained in the following Joint Polar Satellite System (JPSS) documents:

- Joint Polar Satellite System (JPSS) Common Data Format Control Book - External Volume I – Overview
- Joint Polar Satellite System (JPSS) Common Data Format Control Book - External Volume II – RDR Formats;
- Joint Polar Satellite System (JPSS) Common Data Format Control Book - External Volume V – Metadata;
- Joint Polar Satellite System (JPSS) Algorithm Specification Volume II: Data Dictionaries– Block 2.0.0 – Parts 1-9, 28

Information to identify, distinguish and extract all of the X-band unique source packets from the Suomi NPP and JPSS-1 mission data streams is contained in the respective Joint Polar Satellite System (JPSS) Mission Data Format Control Book (MDFCB).

These documents are available via the DRL Web Portal at:
https://directreadout.sci.gsfc.nasa.gov/?id=dspContent&cid=16

9 Additional Command Line Tools

Additional scripting tools in the 'rt-stps/bin' directory provide additional functionality to RT-STPS.

9.1 Getstatus

The getstatus script periodically retrieves status information from the active RT-STPS server on the local host until it is stopped. The format is:

Linux

```
/home> cd /home/username/rt-stps
/home/username/rt-stps>./bin/getstatus.sh
```

Windows

```
C:\Users> cd C:\Users\username\rt-stps
C:\Users\username\rt-stps> bin\getstatus.bat
```

9.2 Load

The load script loads a local configuration file into the local server. The format is:

Linux

```
/home> cd /home/username/rt-stps
/home/username/rt-stps>./bin/load.sh <path-to-config-file>
```

Windows

```
C:\Users> cd C:\Users\username\rt-stps
C:\Users\username\rt-stps> bin\load.bat <path-to-config-file>
```

9.3 Shutdown

The shutdown script halts data processing and unloads the current configuration file on the local server. The format is:

Linux

```
/home> cd /home/username/rt-stps
/home/username/rt-stps>./bin/shutdown.sh
```

When the script completes, the RT-STPS Sender (if used) will display a “Connection reset” pop-up window.

Windows

```
C:\Users> cd C:\Users\username\rt-stps
C:\Users\username\rt-stps> bin\shutdown.bat
```

When the batch file completes, the RT-STPS Sender (if used) will display a “Software caused connection abort: socket write error” pop-up window.

9.4 Version

The version script prints RT-STPS version information and exits. The format is:

Linux

```
/home> cd /home/username/rt-stps
/home/username/rt-stps>./bin/version.sh
```

Windows

```
C:\Users> cd C:\Users\username\rt-stps
C:\Users\username\rt-stps> bin\version.bat
```

9.5 Rate Buffering Program

RT-STPS includes a rate-buffering program (Rat) that can spool data to a slow target (e.g., such as through a slow network). The server may run slowly because all outputs are processed in one control loop, without internal buffering or internal multi-threading. By using the Rat as an intermediate destination, the server can keep one slow target from slowing the outputs of other targets being serviced at the same time.

Rat may be run on the same computer as the server, or from a remote location. It requires three arguments when it is invoked:

Linux:

```
/home> cd /home/username/rt-stps
/home/username/rt-stps>./bin/rat.sh <inputPort> <targetHost> <targetPort> &
```

Windows:

```
C:\Users> cd C:\Users\username\rt-stps
C:\Users\username\rt-stps> bin\rat.bat <inputPort> <targetHost> <targetPort>
```

Rat listens for socket connections on its input port. When a connection is made, it connects to the target on the target port, forming an end-to-end connection. However, it can only service one input connection at a time. Once connections are established, Rat will buffer input data while sending some data to the slower output connection. Once the input connection closes, Rat will continue to send buffered data to the target until it has no more data, at which point it will close the target connection. Rat forms end-to-end connections after the RT-STPS server has connected to it and will run continuously, listening for and accepting connections until it is terminated.

To stop the Rat program, find its process ID and use the appropriate tools/commands to terminate it. For example:

Linux

Search for the correct rat.sh process ID, and terminate it. Use commands such as “ps aux” and “grep” to find the correct “gov.nasa.gsfc.drl.rtstps.clients.spooler.Main” process (the one with the desired input port, target host, and target port). For example:

```
/home/username/rt-stps> ps aux | grep “gov.nasa.gsfc.drl.rtstps.clients.spooler.Main
<inputPort> <targetHost> <targetPort>“
/home/username/rt-stps> kill <Process-ID>
```

(where <inputPort>, <targetHost>, and <targetPort> are the target host and port specified when the rat process was started)

Windows

Search for the javaw.exe process ID associated with rat.bat, and terminate it. Use a command such as “wmic” to display the full command lines and process IDs of all active javaw.exe processes in order to identify the correct javaw.exe process (i.e. the one associated with “gov.nasa.gsfc.drl.rtstps.clients.spooler.Main”), if there are multiple instances running:

```
C:\Users\username\rt-stps>wmic process where name=“javaw.exe” get
ProcessID,CommandLine
C:\Users\username\rt-stps>taskkill /F /PID <Process-ID>
```

10 Additional Server Configuration Options

The RT-STPS server is configured using command line arguments and system property definitions, or through the JSW configuration file on Linux.

The server takes a *name* argument when invoking it. The full server name then becomes “RtStpsServices.” + *name*. The default full server name is “RtStpsServices.A” if no other name is specified.

The server can be configured using several system properties. Set them when using the -D attribute (e.g.,”-Dport=4935”). If not set, default values will be used as described below.

a) -Dconfig=default.xml. A configuration file to be used until overridden by a loaded file. The default name is “default.xml”.

b) -Dport=4935. The port number that the Server reads for telemetry data. The default port is 4935.

c) -DbufferSizeKb=16. The amount of data to accumulate before processing. The default is 16 kb.

d) -Dsetup=./config. The directory where local configuration files are located. If provided, all files must be within the directory tree. The default is ‘rt-stps/config’.

e) -Dlog.stdout. If specified, log messages are written to the standard output.

f) -Dlog.file=<file>. If specified, log messages are written to <file>.

These arguments and system properties are set in the “rt-stps/bin/server.sh” (Linux) or “rt-stps\bin\server.bat” (Windows) files, or in the configuration file “rt-stps/jsw/conf/rt-stps-server.conf” in the JSW on Linux.

If the server fails to initialize properly, and errors occur before the log service is found, some error messages may go to console instead.

10.1 Configuring the Log

If none of the "-Dlog.*" properties are specified in the server's arguments, log messages are written to the "rt-stps/rt-stps-server.log" file, and standard output is displayed directly on the console.

Additional log options are available as follows:

a) Write log events to a file: specify –Dlog.file=<path-to-log-file>, and the file will be created if it does not exist.

b) Write events to stdout: specify –Dlog.stdout, and any event will be written to the standard output.

10.2 Automatic Setup

If the server is running but no configuration file is loaded and data arrives on its incoming data port (port 4935), then the server will use the "default.xml" file as its configuration file.

Similarly, if the server is running but has not been reconfigured with a new configuration file, and data unexpectedly arrive on its incoming data port, then the server will use the previous configuration file.

If the server is shutdown, it will revert to using the "default.xml" file when next started.

11 Alternate Server Interface

There is an additional command interface to the server accessible by software. This interface is useful to integrate the RT-STPS into a larger system with customization (e.g., connect it to scheduling software that may be modified to run RT-STPS).

The interface is through port 5935 by default. The server expects to receive certain text string messages on this port for control. Each message should have the normal line terminator, and case is significant. The messages are commands to load and shut down sessions. The available commands are:

a) loadgo <configurationFileName> - the server will load the configuration file from its configuration directory, and then enables itself for processing (e.g., "loadgo npp.xml");

b) shutdown - stops processing and unloads the current configuration, which closes all output files.

The server does not send responses to any of these commands. The only feedback is via the server's console window. It will print the usual load and shutdown messages. It will also print error messages labeled as "ProxyThread" messages if it encounters them.